Software Design

... and what scrum tells us about time management.

Software Development Stages

- Understanding the problem domain
- Requirements and specification
- Design and architecture
- Implementation
- Testing
- Deployment
- Maintenance

Software Development Stages

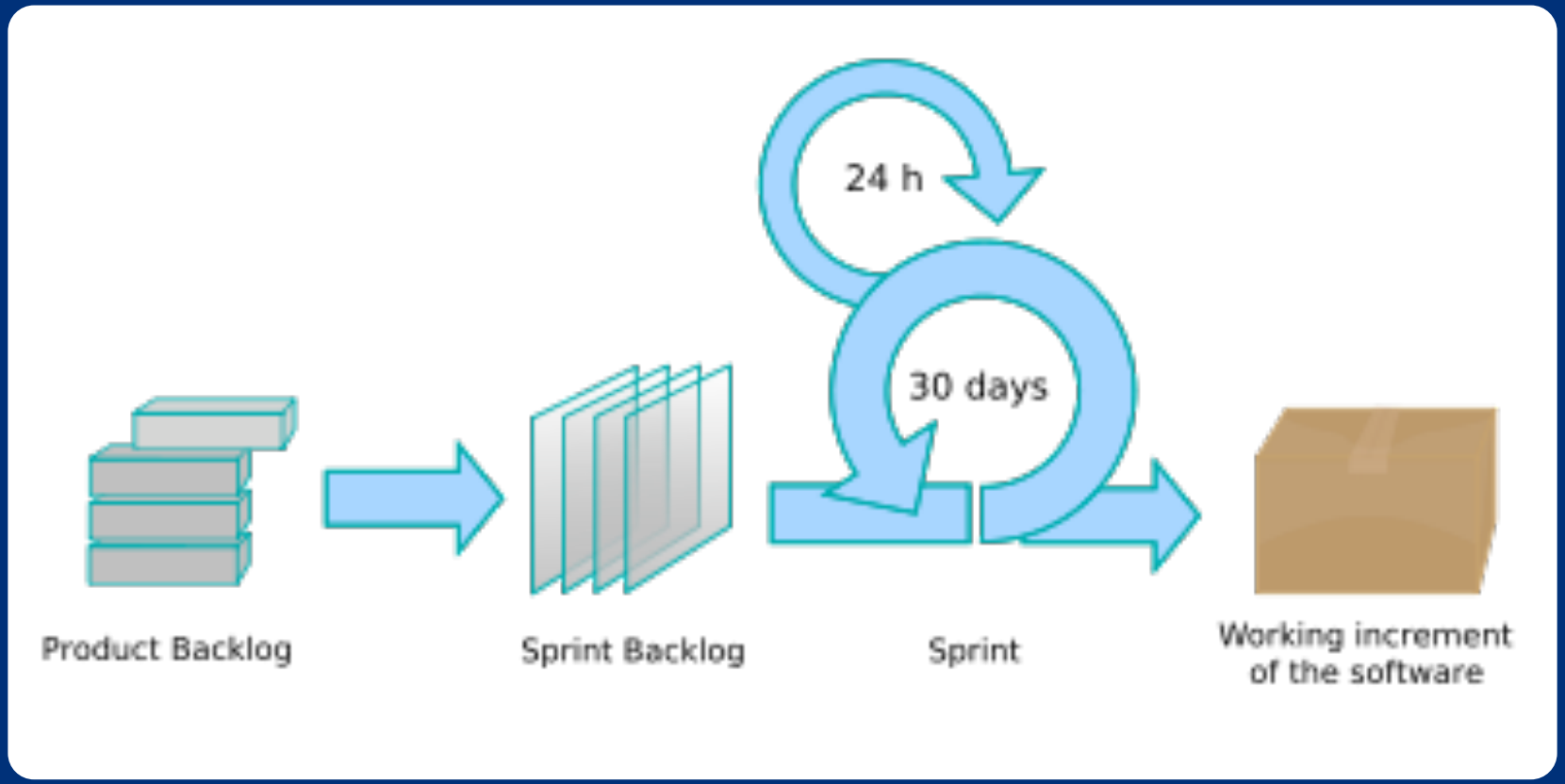
- Understanding the problem domain
- Requirements and specification
- Design and architecture
- Implementation
- Testing
- Deployment
- Maintenance

This stage is critical for defining how the team spends its time.

Key Features of Agile Development

- User stories to drive design
- Lightweight, evolving design as code is written and features are added
- Continuous unit testing
- Pair programming
- Continuous refactoring

A Scrum



- A scrum team works on month-long sprints. In each sprint, the team implements a subset of features selected from a product backlog.

Why Scrum?

- We like to think of development as predictable

  ... but it's not.

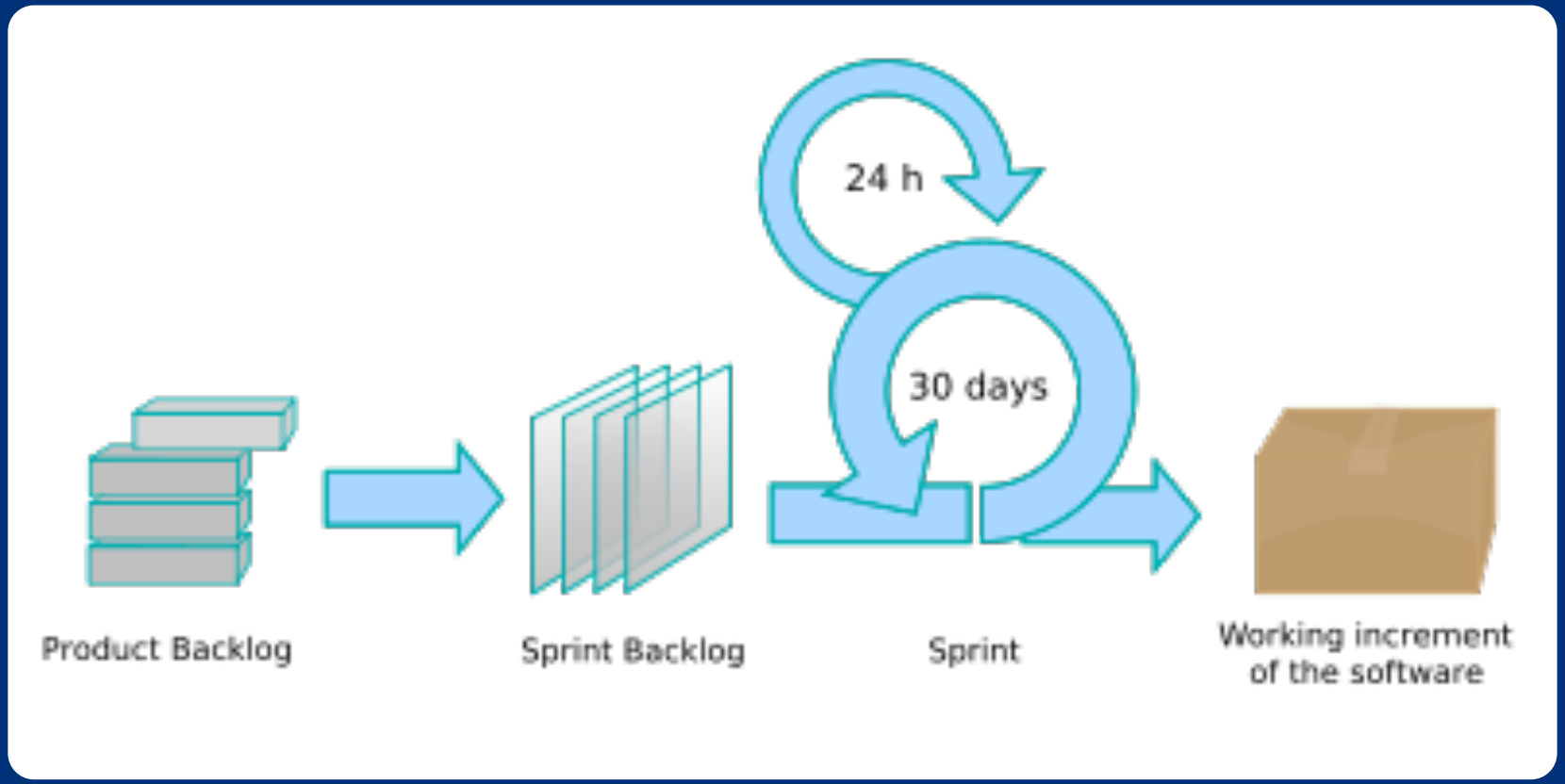
- The customer can change his or her mind.
- Unexpected challenges will arise.
- Some tasks will be easier or harder than anticipated.
- Scrum is designed to make the team deliver quickly and respond flexibly to challenges.

Time Management

- Software design processes can teach us a lot about time management.

- A process is designed to help a team of programmers complete a task within a defined timeframe.
- We can use similar techniques to manage our own tasks -- our own backlogs.
- Scrum is a particularly good one to study, since we need flexibility and the ability to respond to challenges.

A Scrum



- A scrum team works on month-long sprints. In each sprint, the team implements a subset of features selected from a product backlog.

Sprints

- The product owner produces the first version of the product backlog -- a list of features to implement.
- Each cycle, the team chooses which items from the product backlog to move to the sprint backlog. Time estimates are attached to each item.
- During the sprint: daily team meetings (seriously)
- At end of sprint, new features are complete: documented and tested and ready for release.
  - Product increment: something shippable.

To-do Lists

- The product owner (you) produces the first version of the product backlog -- a list of features (tasks) to implement (do).
- Each cycle, the team (you) chooses which items from the product backlog to move to the sprint backlog. Time estimates are attached to each item.
- During the sprint: daily team meetings (seriously)
  - You can always talk to yourself, I guess.
- At end of sprint, new features are complete: documented and tested and ready for release.
  - Your tasks are complete and on time.

Product Backlog

- A list of all tasks needing to be completed.
- The list is in priority order, and each item has a size.
  - The product owner prioritizes this list.

- The backlog contains:
  - bugs, user stories, enhancements, issues, questions
- A to-do list contains:
  - assignments to complete, problems that have arisen, questions to ask, reminders, etc.

Estimating Size

- How big is a feature?
  - Each item has a priority and an estimate of work.
  - Estimates are not in hours, but in relative size!
  - Some people use Fibonacci numbers: 1, 2, 3, 5, 8, 13, 21.
  - Find the smallest item on the backlog and give it a 1, find the largest item on the backlog; give it a 21, and so on.

- Similarly, when writing a to-do list, add sizes and priorities.
  - Don't use time -- that's rarely accurate. Just provide relative priorities.

When is a Feature Finished?

- At the end of each sprint, the tasks in the spring backlog should be finished.
  - Testing must be integrated throughout the sprint.
- The final product should be clean, working, easily extensible code.
- It should have user documentation.
  It should be fully tested.
  In short, it should be ready to hand to a customer.

To Recap

- We've learned a little about a software development process called the "Scrum" process.

- Difficult and high-priority items should be completed first. (Prioritize requirements.)
- Frequent re-evaluation and assessment is important. (Daily team meetings, 24-hour cycles.)
- The backlog contains more than just tasks. (Bugs, questions, and issues are also included.)
- Items should always be "working" (or ready to release).

Software Development Stages

- Understanding the problem domain
- Requirements and specification
- Design and architecture
- Implementation
- Testing
- Deployment
- Maintenance